BUENAS PRÁCTICAS

EN EL TESTING DE SOFTWARE

Quassure

No bugs. Just pure software quality.

1

Para garantizar el éxito de cualquier proyecto, es crucial establecer una estrategia de pruebas sólida que incluya la planificación y el diseño adecuado. Esto implica definir claramente los objetivos del testing, delimitar el alcance de las pruebas y asegurar la asignación adecuada de los recursos necesarios.

2

Integra las actividades de prueba desde las primeras etapas del desarrollo para colaborar, identificar y solucionar problemas de manera temprana, reduciendo así costos y el tiempo dedicado a correcciones posteriores.

3

Fomenta siempre la comunicación entre el equipo de desarrollo y testers. Comparte información y documentación relevante, así como herramientas que faciliten las tareas de ambos equipos. Esta comunicación constante y el intercambio de recursos ayudan a mejorar el desempeño de las tareas, así como a detectar problemas de manera temprana, contribuyendo a la entrega de un software de mayor calidad.

4

Es importante retroalimentarse de cada proyecto realizado para identificar áreas de mejora y corregir las técnicas empleadas en el proceso de pruebas. Esta retroalimentación debe basarse en un análisis exhaustivo de los resultados obtenidos durante las pruebas, permitiendo identificar patrones, tendencias y oportunidades de optimización. Mediante la retroalimentación y el análisis de resultados, se promueve un ciclo de mejora continua.

5

Para alcanzar una mayor cobertura y eficiencia en tus tareas implementa pruebas automatizadas. Deberás identificar aquellos casos que sean adecuados para la automatización y realizar de forma regular el mantenimiento de los scripts.

6

Diseña tus pruebas basándote en el perfil de los usuarios además de aplicar las técnicas sobre los requisitos funcionales. Crea pruebas que reflejen los diferentes perfiles y escenarios de uso reales. Esto ayudará a identificar problemas y mejorar la experiencia del usuario.

7

Implementa pruebas de regresión de manera inteligente. Si no tienes un buen set ya automatizado que te ayude con esta tarea, debes utilizar tus conocimientos técnicos para identificar las áreas de mayor riesgo y priorizar las pruebas correspondientes a ejecutar. Esto te ahorrará tiempo además de darte a ti y a tu equipo tranquilidad de que has realizado una cobertura adecuada de las funcionalidades críticas.

8

Utiliza y combina técnica de caja negra y caja blanca para así tener una mayor cobertura tanto en extensión como en profundidad. Enfócate tanto en la interfaz y el comportamiento externo como en la lógica interna y estructura del código.